BBIPED: BCAM-Baltogar Industrial Platform for Engineering Design

C. Alonso-Montes, I. Garcia, A. Ramezani, L. Remaki

Abstract—Currently, commercial software for computational fluid dynamics offers a good set of features to deal with traditional designs. Within a competitive market industrial innovation is a key factor that must be faced by companies. However, the design of solutions to deal with industrial challenges cannot be done within commercial software due to the lack of flexibility. Open source initiatives are a good option but the learning curve is high, specially for industrial engineers profiles. In this paper, we present the BBIPED platform which has been designed to deal with turbomachinery applications in a simple and friendly way. The main goal is to keep the platform as simpler as possible providing the enough flexibility to include out-of-the-box solutions to cope with industrial challenges. BBIPED platform provides links with currently existing remarkable open source initiatives altogether with our own developments. Particularly, it is remarkable a first approach for automatic mesh generation based on geometry parametrization solution, and the provision of novel techniques to deal with multiple rotating frame (MRF): Multizone MRF an Virtual MRF. Case tests were designed to test the solutions and to assess and validate the results against commercial suites with promising results.

Keywords—CFD, Virtual Rotating Frame, Multizone Rotating Frame, Automatic mesh parametrization

I. Introduction

COMPUTATIONAL FLUID DYNAMICS (CFD) is a key area extensively used in a wide range of industrial processes in domains like aeronautics, automotive, ventilation, chemical manufacturing, oil industry, power generation, etc. Commercial software tools are currently available in the market that cope with the whole **CFD workflow** (meshing, solver simulation, data post-processing) to deal with traditional industrial problems, providing a good set of powerful and robust solutions. However, these tools lacks the flexibility to design out-of-the-box CFD solutions to deal with new scientific industrial challenges. In this sense, open-source alternatives provides enough flexibility for engineers to define their own and customised solutions. Some of them have gained a strong reputation due to its robustness, accuracy and flexibility within the scientific community, such as: Salome Platform [1] for CAD and mesh generator, SU^2 [2], [3] or FeNICS [4] for solver engines, or Paraview [5] for data postprocessing. One of the biggest open source tool players is OpenFoam [6], which covers the full CFD workflow providing flexible solutions for a wide range of industrial domains. Although the great performance and flexibility of OpenFoam, its learning curve for engineers is huge, specially in terms of the tailoring it for novel solutions within the platform.

In this context, and particularly focusing on turbomachinery applications, there is a need of an open-source solution capable of reducing the learning step for industrial engineers, and maintaining the flexibility to add new CFD solutions. Following this approach and with the collaboration of Baltogar S.A, the **BCAM-Baltogar Industrial Platform for Engineering Design (BBIPED)** platform was designed to cope with the full CFD workflow for turbomachinery applications; from CAD and mesh design and generation, solver simulation and data post-processing. BBIPED design aims to take advantage of the integration of existing open-source cutting edge technologies together with new module developments, fulfilling industrial needs in terms of accuracy and performance. BBIPED platform uses the following open-source initiatives: Salome Platform for (CAD/mesh generation), SU^2 for CFD solver simulation, and Paraview for data post-processing and visualization. All of them are integrated through a common interface that will help the engineers to easily set up the working environment. BBIPED platform offers also novel features and functionalities tailored to turbomachinery applications, e.g. automatic geometric adjustments and generation for specific fan design, specific developments to tackle with the Multiple rotating frame method (multi zone and virtual multiple rotating frames). The main aim of the platform is to ease the usage of these applications within turbomachinery applications, with special emphasis on the final end users.

In this paper, the BBIPED platform will be explained in detail, specially focusing on the novel modules. In Section II the BBIPED platform is presented from a technical perspective, with special emphasize on the virtual rotating and multi-zone approaches, as well as the automatic geometry generation; in Section III some experiments are shown in order to demonstrate the usability of the platform; and in Section IV some conclusions and the future work are presented.

II. BBIPED in a nutshell

BCAM and Baltogar joined efforts to overcome the gap among industrial engineer needs and the usage of open-source initiatives tailored to turbomachinery applications. BBIPED is a fluid dynamic simulator aimed to handle efficiently challenging industrial fluid based applications. The main goal is to take advantage of existing technologies, and to provide new solutions for challenging industrial problems. In this section, the BBIPED platform will be presented, and the main innovative modules will be explained in detail.

The authors are members of the Computational Fluid Dynamics Group, Basque Center for Applied Mathematics - BCAM. SPAIN
{calonso,igarcia,aramezani,lremaki}@bcamath.org

This work was partly funded by the Department of Economic Promotion of the Biscay Foral Council (BFA/DFB - 6/12/TK/2012/00020), through the funding plan for Research Projects in Excellence Centres 2012

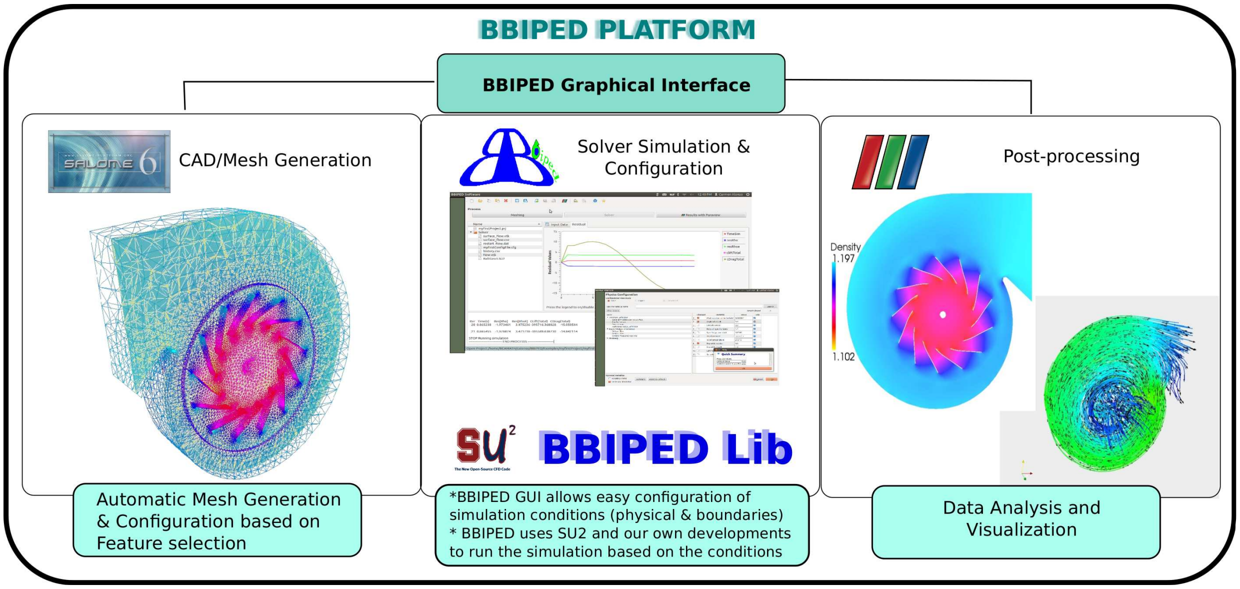


Fig. 1. BBIPED Platform Schema. The BBIPED Platform is composed by 3 main blocks (from left to right): CAD/Mesh generation, Solver configuration and simulation, and Post-processing. Each of the blocks are integrated through a common interface, the so-called BBIPED Graphical Interface(BBIPED GUI).

A. The BBIPED platform

BBIPED platform is conceptually split into three main blocks integrated through a common and user-friendly graphical interface (BBIPED GUI) (see Fig. 1):

- CAD and mesh generation
- Solver Simulation
- Data Postprocessing and Visualization

There are well-known and reputable open source initiatives that could cope with the requirements of the different blocks. So, BBIPED platform has been designed to integrate those valuable initiatives and giving room for easily adding ad-hoc initiatives. A first selection of open source initiatives have been integrated within the first version of BBIPED.

- The CAD and mesh generation will be managed through Salome platform, since there is the possibility to provide the flexibility to add customised functionality for automatic geometric parametrization. In BBIPED, some automation for specific geometry parametrization is provided (see subsection II-C for further details).
- Solver simulation is based on SU^2 tool suite. Customized developments regarding multizone rotating approaches are also provided by BBIPED site (see subsection II-D for further details)
- For the data post-processing and visualization, the Paraview tool was the best option for our needs.

In any case, the user has the possibility to change the CAD/Mesh tool as well as the data post-processors. Notice that OpenFoam was not selected, due to its size, complexity and generic nature. Indeed, tailoring OpenFoam to our requirements would be expensive in terms of cost and time. Furthermore, the learning curve for industrial engineers will be quite big.

A graphical user interface (GUI) is essential to ease industrial adoption of any software tool suit. In this sense, the BBIPED platform offers their own GUI (BBIPED GUI, see Fig. 2). This GUI has been designed to ease the usage by the engineers by means of the promotion of a standardisation and unification of the configuration process for the solver simulation. This GUI is offered to supply the lack of a SU^2 GUI, and to integrate our own conditions to run BBIPED Lib modules. In this sense, any change in the Solver engine will be transparent for the user

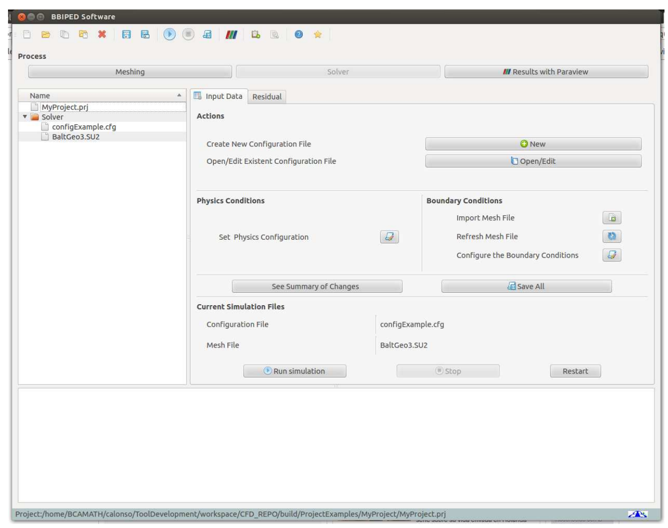
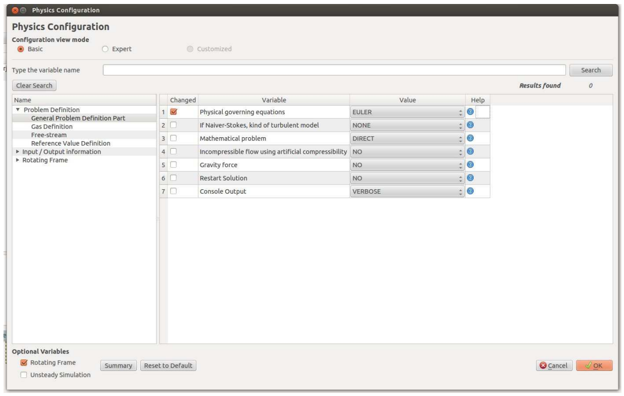


Fig. 2. BBIPED Graphical User Interface (GUI) example. The solver configuration is made and controlled through this window.

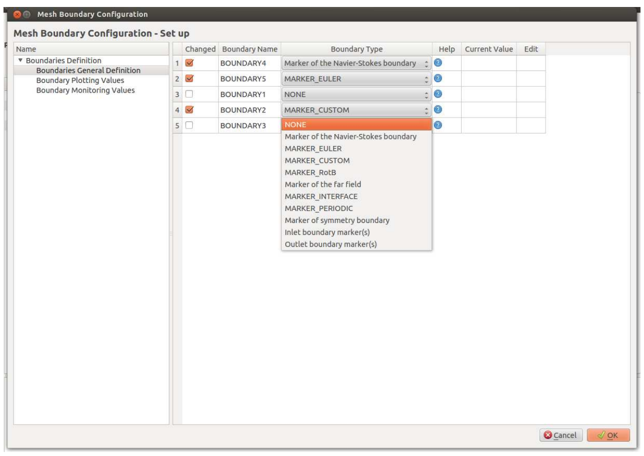
B. BBIPED Graphical User Interface

The main aim of the BBIPED GUI is to offer an automated and standardized CFD workflow to better manage the whole process from CAD generation, meshing, solving, and post-processing. Its main features are:

- Mesh Creation based on Salome Open Source platform, launched through BBIPED GUI



(a)



(b)

Fig. 3. Configuration of the simulation conditions: 3(a) helps in the definition of physical conditions and 3(b) helps in the boundary conditions associated to the mesh

- Case Tailored templates for automatic CAD and mesh generation for specific turbomachinery cases
- Innovative virtual multiple rotating frame and multi-domain approaches implemented to cope with industrial needs (included in BBIPED lib)
- Post-processing for data visualization launching Paraview tool

For mesh generation, Salome Platform has been considered from the very beginning. Some facilities to automate the mesh generation are also provided with the GUI, see II-C for further details.

For solver simulation, the main solver engine tested was SU^2. One of the drawbacks of using SU^2 as a solver engine is the lack of graphical interface. To overcome this issue, the BBIPED GUI has been designed to offer solver configuration capabilities compatible with SU^2. Moreover, the new functionalities provided by the BBIPED Lib, are smoothly integrated within the BBIPED GUI. This GUI has been developed using Qt technology [7] in order to easily integrate C++ developments, c++ based libraries and the possibility of OS-cross platform development. Currently, the first release has been fully tested in Ubuntu 12.04 and 13.0. In order to run any simulation, the main physical and boundary conditions must be established. BBIPED GUI offers the possibility to independently configure from one side the physical conditions and from other side, the boundary conditions. The **physical conditions menu** (see Fig. 3(a)) are those conditions related with the solver and physical equations needed for the CFD simulation. In this case, BBIPED GUI offers different operating modes depending on the expertise of the engineer:*basic, advanced or personalized configuration view mode*. The basic one shows only those values to be configured for engineers with no expert knowledge in solver equations, setting some values by default. The advanced view shows all the variables but it is recommended only for real experts. The personalized view shows only a subset of variables to be configured. This is specially oriented for those projects where only a small set of variables need to be configured. The **boundary conditions menu** (see Fig. 3(b)) allows to the engineer to configure the boundaries of the correspondent mesh. The boundary mesh names are automatically extracted from the mesh file, avoiding naming errors and focusing the engineering work in the boundary set up. From the configuration done in both physical and boundary menus, an unique configuration file is obtained, which is also a SU^2 compliant. Notice that customized solver engines could be also used to run the same project, but they must be SU^2 format compliant. The evolution of the simulation and the residuals could be checked in two ways, numerically from the BBIPED GUI Console or graphically.

C. Geometry Parametrization and Automatic Mesh Generation

Usually, industrial applications uses variants of similar geometries or meshes of their products. However, it is a real challenge to provide a general automatic geometry generator. In BBIPED, we provided some specific automatic mesh generation based on geometry parametrization. We defined a formal procedure based on parametrization points and rules (see Fig. 4).

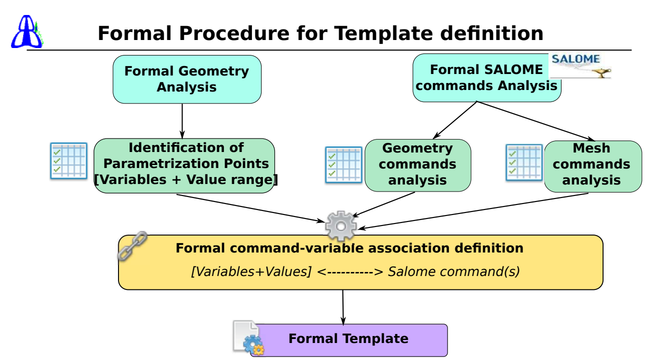


Fig. 4. Formal procedure to manually build up the templates for automatic geometry generation

Firstly, a formal and deep analysis of the geometry must be performed in order to define and identify the appropriate parametrization rules that are governing the different variants. However, this process is completely manual. In our case, and after analysing industrial fans used at Baltogar, suitable parameters were extracted to completely define a set of similar industrial fans in terms of bald curvature, bald number, casing shape, etc. These terms are consider as the parametrization points, formed by a variable name and a range or set of feasible values. Notice that these parametrization points will be used later on by the engineer to define the different variants of the mesh.

One of the best features of Salome platform is the possibility to define customized programs for mesh generation (in Python language). So, once the parametrization points and the formal analysis of the mesh has been performed, the identification of the Salome commands must be done. This step needs an expert knowledge on Salome platform. Once the set of parametrization points are identified and the Salome commands are defined, the engineer must define the appropriate template command file to build up the geometry associating the commands with the correspondent variables. This will conform a formal template. The formal template will serve as the basis for building up the geometry according the correspondent variables and values. This formal template is generic for the specific set of variables. Currently, a set of already predefined templates is provided for the generation of industrial fans.

Once the formal template is defined, BBIPED GUI allows an easy and user friendly way to use them by the engineers. BBIPED GUI can load the templates and the set of predefined parametrization points. The final user can configure the desired values for each of the variables (parametrization points). Then, the BBIPED geometry engine will build a specific Salome compliant geometry file based on the user input and the associated formal template to build the final mesh in an automatic way. The full process is shown in Fig. 5.

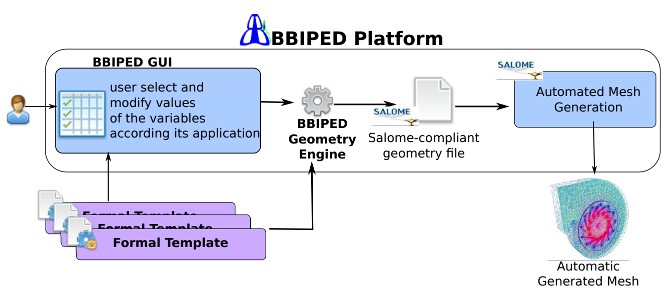


Fig. 5. BBIPED Geometry Automated Schema: showing how the template system works

The presented functionality is worthy for those situations where variants of the same geometry need to be produced, speeding up the initial mesh generation. Furthermore, new variants of the same geometry could be easily adopted by the company based on the analysis already made. This approach forces an standardised mesh generation reducing human errors and speeding up the mesh creation.

D. Multiple Rotating Frame

Realistic flow simulation is a key success factor as the basis of optimal design in turbomachinery. Indeed, the accurate and cost effective simulation of vortex flow caused by fixed and rotating frames interactions is one of the biggest challenges in turbomachinery [8], [9]. One of the most popular methods used to simulate the rotating effect is the so called **Multiple Rotating Frame (MRF)** introduced by [3], [10]. This technique consists in the creation of an interface among rotating and static parts of the geometry, splitting the geometry into several zones independently meshed that communicate through **interfaces** by variables or fluxes exchange.

One of the most interesting features of MRF is that the unsteady flow could be solved as an steady state problem. Notice that due to the presence of fixed (volute) and rotating parts (rotor) simultaneously in the domain, this domain is unsteady (the flow in the domain is time dependent). But, in the MRF schema, the physics are viewed by the observer who is attached to the rotating frame [11] and on that view the flow is steady. So, the interface vertices belong to two different zones: rotating and stationary. The control volume for duplicated vertices (ones on the interface) should be the same, resulting in an automatic matching of the solution of both zones. Notice that in this approach it is not guaranteed that in both zones the flow properties on these common nodes to be identical. The Navier-Stokes equations, solved in a relative frame (associated to the rotating part), have just some extra terms to consider constant angular velocity of the domain. In each iteration, the information is transferred through the interfaces; so, the scheme which is applied for is important. Besides, since physically on the interface the velocity should has minimum radial velocity, the interface definition, which is the outer boundary of the rotating part, is crucial since any misplacing may cause divergence or non physical phenomena [12]. Another advantage of the MRF technique is the preservation of the autonomy in every zone. Furthermore, it is also suitable for zones that have common faces but not necessarily common nodes, like in sliding meshes or mixing planes or in some over-set grids approaches.

One of its main drawbacks is its inherent technical complexity in terms of geometry and from numerical simulation perspective. For the geometrical point of view, the interface must be created at CAD-level, which complexity grows specially in those with strongly connected rotating and static parts. Furthermore, the interface definition at CAD-level must be supported by a suitable mesh tool capable to build a grid with more than one closed domain and this could not be achieved by a mesher using a simple Delaunay strategy for instance. Even if nowadays modern meshers could achieve such a task, it is still not the case for many simple mesh generators. Finally, appropriate boundary conditions should be defined in the solver level to ensure adequate transfer of the information between the rotating and stationary domains. In terms of numerical simulation, two different set of equations (in stationary and rotating frame) are solved at both sides of the interface, so its position will affect the results and convergence [13]. In this sense, both the optimal position and the boundary condition at the interface must be defined a priori. Indeed, the optimal position cannot be accurately defined prior the simulation. So, its adjustment must be done at CAD level, re-meshed and simulated again.

SU^2 implements rotating frames and some basic multi-zone facility. In BBIPED platform, a new implementation has been included following the classical MRF technique which

ISBN: 978-1-61804-241-5

special consideration on the data transfer techniques at the interface, the so-called **Multi-zone-MRF approach (MMRF)**. The whole description and motivation is described in [14]. In the MMRF approach, a user can set an arbitrary number of rotating and static parts and the correspondent number of meshes in a single simulation. So, the user can have in a single simulation a complete view of the residuals.

To overcome some of the drawbacks of MRF, in [15] a simplified approach of MRF was proposed, the **Virtual Multiple Rotating Frame(VMRF)**. In this technique, a virtual interface among rotating and stationary parts is built at the solver level, defining a virtual axisymmetric zone containing the rotating domain. The virtual zone can be made by a revolving curve which makes it easy to identify rotating nodes. Complex interfaces could be obtained by assembling simple geometries like cylinders, cones, spheres and so on. In this approach, the virtual interface is easily identified by the edges that have one vertex in the rotating zone (cylinder) and the other outside, which are the ones that actually need a specific treatment.

Both techniques were numerically tested against commercial software with promising results, as it is shown in [15]. Both methods are integrated into the BBIPED platform (within the BBIPED Lib), providing a formal procedure with both methods which eases the learning step for engineers. In one hand, VMRF projects are treated as traditional BBIPED projects, this is, the engineer will define the correspondent boundary and physical conditions, and then the specific rotating features. The BBIPED GUI automatically set up the environment to proceed with rotating features. On the other hand, BBIPED GUI offers the possibility to create specific multi-zone projects to handle MMRF approach. This Multi-zone facility prepares the working environment for the multi-zone execution, keeping the same configuration step workflow. In this case, the engineer will need to individually define a mesh for each zone with their corresponding physical and boundaries conditions following the traditional project structure. This philosophy unifies the treatment of the multi-zone mesh avoiding to add new complexity to this step.

III. Experiments

As it has been previously commented, BBIPED Platform has been tailored to turbomachinery applications. BBIPED GUI has been designed using Qt Technology developed under Linux environment (tested on Ubuntu 12.04 LTS and Ubuntu 13.0). Initially, only the linux version has been delivered, although it is expected to release for other OS, on demand. The BBIPED Lib including the developments for MMRF and VMRF have been developed using C++ and SU^2 version 2.3 as a library. Currently, we are migrating the developments towards the new version of SU^2 version 3.0. The developments for the automatic mesh generation approach has been developed using Python and the correspondent API of Salome Platform. In this section some practical examples will be shown for the main modules presented before: *automatic mesh generation, MMRF and VMRF*.

A. Automatic mesh generation approach: An example

Automatic mesh generation based on geometry parametrization is a current challenge in industry. We analysed several fan geometries from Baltogar to extract the parametrization rules and points. As a matter of example, some of the parametrization points identified are the following: number of blades, blade stagger angle,blade leading edge radius, etc. All of them characterise the full set of different variants of Baltogar fan geometries. Once the set of parametrization points are identified, the correspondent Salome commands must be identified. An example of the Salome command for the blade number selection is shown:

```
Multi_Rotation_1 =
geompy.MultiRotate1DNbTimes
(Blade_Base_Surface,OZ,"Blade_Number")
```

The engineer will select through the BBIPED GUI the number of desired blades for the fan, and the BBIPED platform (through Salome) will automatically generate the mesh with the correspondent number of blades (see Fig. 6).

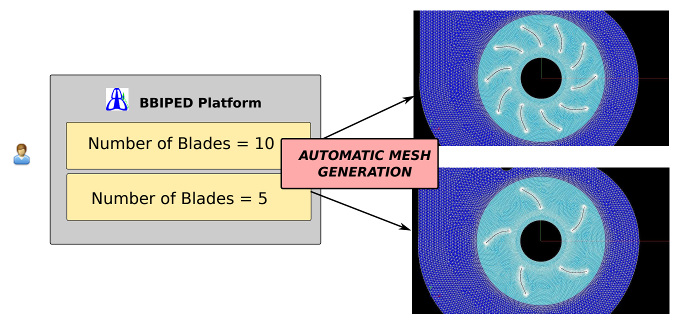


Fig. 6. Example of automatic generation of a fan with different number of blades. In one case, 5 are selected and the final mesh will contain 5 blades, whereas 10 is selected, it implies the generation of the correspondent 10 blades. Notice, that only the number of blades are needed and the mesh is automatically generated in accordance

B. MMRF and VMRF: examples

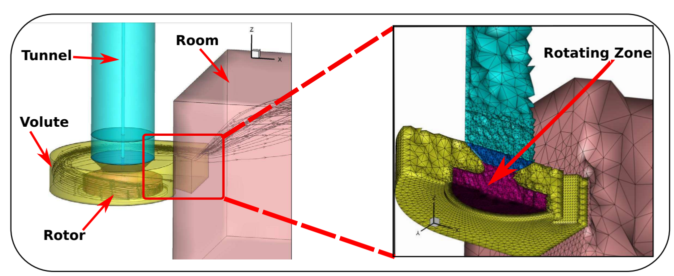


Fig. 7. Multi-zone Rotating Frame example. On the left, the original domain is shown: a fan to extract the air from a tunnel to another room. On the right side, a section of the mesh is shown. Notice that the fan has a rotating part (rotor) inside a static one (volute)

MMRF and VMRF approaches have been shown as powerful alternatives to deal with rotating and static elements within the same domain. Both of them are available within the BBIPED platform. In this section, we are going to show the usage of both within the platform through the example

ISBN: 978-1-61804-241-5

proposed in Fig. 7. In this example, the fan extracts the air from the tunnel into the volute (yellow part) through the rotating rotor (orange part) towards the room. Each of the domain elements is discretized and every zone must be clearly defined through the different meshes. Notice that physically each element of the domain (tunnel, room, volute and rotor) has their own physical mesh.

In this particular example, for the MMRF approach needs so many meshes as areas explicitly indicating the interface between them. However, the VMRF approach allows the engineers to identify which are the interface areas within a global mesh without explicitly providing the physical mesh. BBIPED platform can handle both approaches. In Fig. 8, it is shown the different procedure in both cases within the platform. The MMRF needs as much meshes as parts are considered, in the example for the rotor and for the volute. This must be handled at the Mesh generation step (this is in Salome platform). Then, the BBIPED GUI provides a mechanism to configure each mesh independently. The simulation is performed simultaneously over all the meshes and the configuration files. In the case of the VMRF, only one mesh is actually needed. This implies that the engineer needs a single configuration for the whole domain, just indicating the rotating parts.

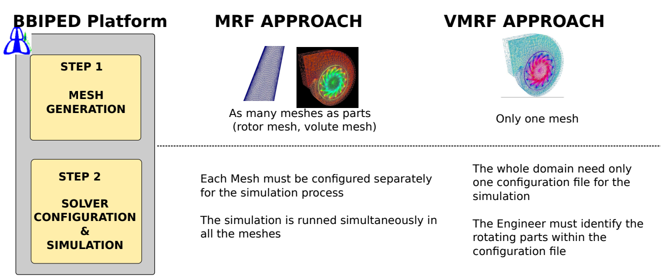


Fig. 8. Handling MMRF and VMRF approaches within BBIPED platform

Both methods, MMRF and VMRF from the BBIPED platform were assessed and validated against commercial tools to assure meeting industrial accuracy requirements. The assessment has been performed through the specific design of a 2D case, including all basic aspects of turbomachinery cases including rotating fan, fix volute and mixing zone with high speed free stream (see Fig. 9). This case has been solved using BBIPED platform and the commercial platform.

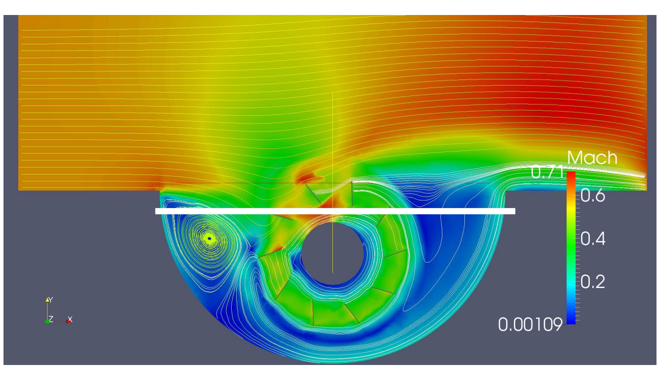
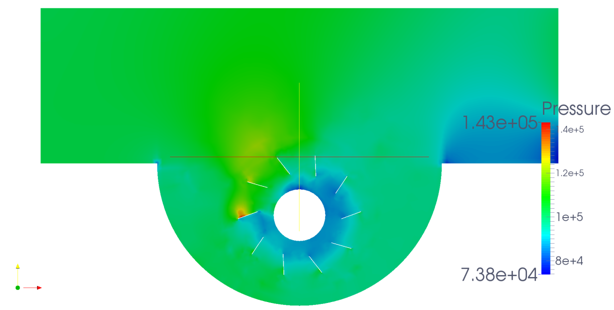


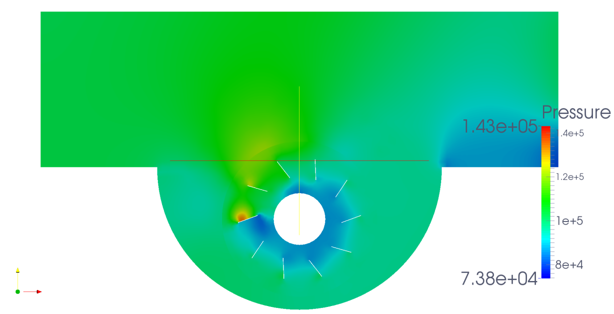
Fig. 9. Mach contours and streamlines

A 0.6 Mach flow through a fan with an angular speed of 150 rad/s is simulated. The flow is assumed to be inviscid and compressible. The domain is discretized by a full unstructured quads. This case has been solved respectively by VMRF, MMRF and Ansys-Fluent v. 14.5 (using MRF) [16]. Due to the interaction of the free stream and the rotating fan, a complex flow is observed which is depicted using streamlines and Mach contours in Fig. 9.

The pressure contours of VRMF and Fluent can be compared visually in Fig. 10. For a quantitative comparison, the pressure distribution is extracted from a horizontal line (horizontal white line in Fig. 9), then it is plotted in Fig. 11. These figures show a good agreement of both VMRF and MMRF with the highly reliable commercial software.



(a)



(b)

Fig. 10. Pressure Contours Comparison: 10(a) using BBIPED platform with VMRF, 10(b) using FLUENT

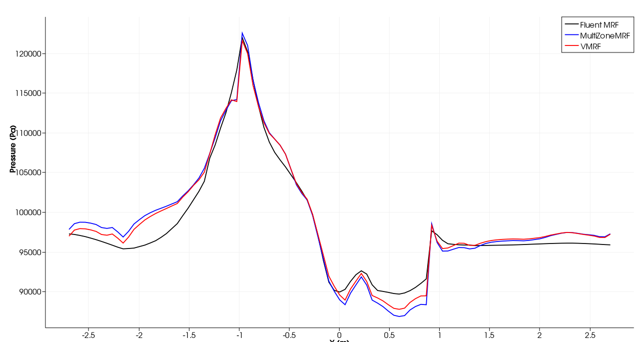


Fig. 11. Pressure values comparison along an arbitrary line among the MRF version of Fluent (black line), and the BBIPED approaches (MMRF -blue line- and VMRF-red line-)

IV. Conclusions and Future work

BBIPED platform has been designed and tested by industrial engineers with real fan cases to analyse weakness and new features to tailor this solution to turbomachinery applications. The set of new features is a potential innovation worthy for real industrial cases. The automatic mesh generation helps in speed up the mesh generation process, as well as reduces the human errors. Moreover, the manual step of analysing geometries could be seen as a good exercise to standardise the geometry design within the company. The usage of SU^2 as a solver allows BBIPED to be used as a valid GUI for a wide range of applications on top of the solver engine. The integration of our own solutions (MMRF and VMRF) was easily integrated through the interface and with the SU^2 tool. This process is transparent for the end user. So, if the user can use different SU^2 compliant solver engines, through BBIPED GUI. Moreover, we have tested the results obtained using the MMRF and VMRF approach against commercial software to validate the accuracy levels, with very promising results. The main aim of the whole platform is to offer the complete work-flow for CFD processes keeping as much simple as possible, to ease the early adoption by industry.

The BBIPED platform is currently under test at Baltogar company. A public version will be released to the general public by the end of the project, in June 2014. Future developments are envisaged to extend the capabilities of the platform to new domains and facilities: turbulence models, vortex filaments, medicine simulations, etc.

Acknowledgment

The authors gratefully acknowledge the financial support of Diputación Foral de Bizkaia (DFB) for this research and the whole BCAM-BALTOGAR project on turbomachinery (grant BFA/DFB-6/12/TK/2012/00020).

References

[1] "Salome Platform." http://www.salome-platform.org/, 2014.

[2] F. Palacios, T. D. Economon, A. C. Aranake, S. R. Copeland, A. K. Lonkar, T. W. Lukaczyk, D. E. Manosalvas, K. R. Naik, A. S. Padron, B. Tracey, A. Variyar, and J. J. Alonso, "Stanford University Unstructured (SU2): Open-source analysis and design technology for turbulent flows," in *AIAA Paper*, 2014.

[3] F. Palacios, M. R. Colonno, A. C. Aranake, A. Campos, S. R. Copeland, T. D. Economon, A. K. Lonkar, T. W. Lukaczyk, T. W. R. Taylor, and J. J. Alonso, "Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design," in *AIAA Paper*, AIAA, 2013.

[4] "FeNICS tool." http://fenicsproject.org/.

[5] "Paraview Tool." http://paraview.org/, 2014.

[6] "OpenFoam Tool." http://www.openfoam.com/, 2014.

[7] "Qt technology." http://qt.digia.com/, 2014.

[8] C. R. Meng Wanli, "Study of helicopter autorotation landing following engine failure based on a six-degree-of-freedom rigid-body dynamic model," *Chinese Journal of Aeronautics*, vol. 26, no. 6, p. 1380, 2013.

[9] A. Califano and S. Steen, "Analysis of different propeller ventilation mechanisms by means of rans simulations," in *First International Symposium on Marine Propulsorsmp*, vol. 9, 2009.

[10] J. Y. Luio, A. D. Gosman, and I. R. Issa, "Prediction of impeller-induced flows in mixing vessels using multiple frames of reference," *Inst. Chem. Eng. Sympo Ser.*, vol. 136, pp. 549–556, 1994.

[11] J. D. Denton, "Some limitations of turbomachinery cfd," in *ASME Turbo Expo 2010: Power for Land, Sea, and Air*, vol. 7, 2010.

[12] Z. Liu and D. L. Hill, "Issues surrounding multiple frames of reference models for turbo compressor applications," in *Int. Compressor Eng. Conf.*, vol. 1369, 2000.

[13] M. Zadravec, S. Basic, and M. Hribersek., "The influence of rotating domain size in a rotating frame of reference approach for simulation of rotating impeller in a mixing vessel," *Journal of Engineering Science and Technology*, vol. 2, no. 2, pp. 126–138, 2007.

[14] A. Ramezani, L. Remaki, and J. A. Suarez, "Bbiped an su2-based open-source: New multiple rotating frame developments," in *Open Source CFD Int. Conf*, 2013.

[15] L. Remaki, A. Ramezani, J. M. Blanco, and J. Antolin, "Efficient rotating frame simulation in turbomachinery," in *Proc. ASME Turbo Expo 2014: Turbine Technical Conference and Exposition (To Appear)*, 2014.

[16] "Ansys fluent: Computational fluid dynamics simulator." http://www.ansys.com/Products/Simulation+Technology/Fluid+Dynamics/Fluid+Dynamics+Products/ANSYS+Fluent, 2014.

Carmen Alonso-Montes Carmen Alonso-Montes obtained her PhD. in Computer Science and Artificial Intelligence at the Univ. A Corua (Spain) in 2008. She worked in Tecnalia Research and Innovation in more than 20 National and European projects related with embedded systems, decision-making systems, and robotics. Then, she was project leader of machine vision projects in Ik4-Ideko for machine tool image-based applications. Currently, she is a post-doctoral fellow at BCAM, in the Computational Fluid Dynamics group.

Imanol Garcia Imanol Garcia obtained the BSc. Chemical Engineer (Univ. of Basque Country, Spain) and the MSc. Process System Engineering degree (Cranfield Univ, UK), both in 2011. He hold a MSc in Research in Mathematics (Univ. Basque Country, Spain) since 2012. Currently, he is doing his PhD on CFD at BCAM, supervised by Professor Lakhdar Remaki. His main research interests are: aero-acoustics and mesh generation techniques for turbomachinery problems.

Ali Ramezani Ali Ramezani obtained his PhD in Aerospace Engineering. He is a CFD specialist. His interests and experiences are focused on numerical methods including: Computational Fluid Dynamics, grid generation, moving boundary problems, parallel processing and dynamic load balancing, Multi-grid, turbulence modelling, overset grids, heat transfer, Finite Volume, Finite Element, Discontinuous Galerkin and Finite Difference, etc. He is a post-doctoral fellow at BCAM, in the Computational Fluid Dynamics group.

Lakhdar Remaki Lakhdar Remaki is a BCAM-BALTOGAR research professor and CFD group leader at Basque Centre of Applied Mathematics (BCAM). He received his PhD degree in Applied Mathematics from Claude Bernard University of Lyon, France. He has been a post-doctoral, research associate and research officer at Ecole Polytechnique of Montreal (Canada), McGill University (Canada), Swansea University (UK) and Sherbrooke University (Canada) respectively. His research expertise include computational fluid dynamic, PDEs, finite element and finite volume methods, shock detection, aerodynamic, gas-particle interaction, acoustics, mesh generation and adaptivity, signal and image analysis. He made several scientific contributions to the different fields published in over 60 journals and proceedings.